'07-20-05

AF IFW

Atty. Docket Nbr. RSW9-2000-0054-US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of Donald J. Boulia

Serial Nbr:	09/619,178	Filed:	July 19, 2000
For:	Technique for Sending TCP Messages through HTTP Systems		
Art Unit:	2155	Examiner:	Oanh L. Duong

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF IN SUPPORT OF APPEAL FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS

Sir:

Applicant herewith submits an appeal brief in support of the appeal to the Board of Appeals from the decision dated March 25, 2005 of the Primary Examiner finally rejecting all Claims 1 - 33.

The appeal brief fee of $500.00 is:

□ Enclosed.
□ Not required. (Fee paid in prior appeal.)
☒ Charged to Deposit Account No. **09/0461**. A duplicate copy of this sheet is enclosed.

Oral Hearing is:

☒ Not requested.
□ Requested. See first paragraph of accompanying appeal brief.

Date: July 19, 2005

Respectfully submitted,

By Marcia L. Doubet
Marcia L. Doubet, Attorney of Record
Registration No. 40,999

Customer Number 43168
Telephone: 407-343-7586
Fax: 407-343-7587

Atty. Docket No. RSW9-2000-0054-US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of Donald J. Boulia July 22, 2005

Serial Nbr: 09/619,178

Filed: July 19, 2000

For: Technique for Sending TCP Messages through HTTP Systems

Art Unit: 2155

Examiner: Oanh L. Duong

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

EXPRESS MAIL CERTIFICATE

"Express Mail" label number: **ER635733802US** Date of Deposit: July 19, 2005

WE REQUEST THE DATE OF DEPOSIT AS THE DATE FILED.

I hereby certify that the following enclosed or attached papers and fee

(1) Transmittal of Patent Appeal Brief, requesting payment of Appeal Brief Fee from deposit account (1 page, in duplicate)
(2) Appeal Brief (42 pages)
(3) Business Reply Postcard

are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on the date indicated above and are addressed to Mail Stop Appeal Brief - Patent, Commissioner for Patents, Alexandria, VA 22313-1450.

__Marcia L. Doubet______________ Marcia L. Doubet
(Name of person mailing paper or fee) (Signature of person mailing paper or fee)

Attorney Docket RSW9-2000-0054-US1

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

In re application of Donald J. Boulia July 19, 2005

Serial Nbr: 09/619,178

Filed: July 19, 2000

For: Technique for Sending TCP Messages through HTTP Systems

Art Unit: 2155

Examiner: Oanh L. Duong

APPELLANT'S BRIEF ON APPEAL

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal seeking reversal of the decision of the Primary Examiner, finally rejecting all current claims of the subject patent application.

1) REAL PARTY IN INTEREST

The real party in interest is the Assignee, International Business Machines Corporation ("IBM").

2) RELATED APPEALS AND INTERFERENCES

Appellant, the Appellant's legal representative, and the assignee, have no personal knowledge of any other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

3) STATUS OF CLAIMS

Claims 1 - 31 stand rejected. Claims 1 - 31 are under appeal.

4) STATUS OF AMENDMENTS

No amendments were filed after the Final Rejection mailed on March 25, 2005.

5) SUMMARY OF CLAIMED SUBJECT MATTER

1. Appellant's independent Claims 1, 10, 19, 28, and 31 specify a "send channel" and a "receive channel" which are distinct (Claim 1, lines 4 - 9; **Fig. 3, 330** and **340**). Appellant's independent claims further specify limitations pertaining to using these distinct channels to provide support for bi-directional protocols (such as TCP, as specified in Claims 1, 10, and 19) over uni-directional protocol systems (such as HTTP, as specified in Claims 1, 10, and 19); specification, p. 7, lines 15 - 16. For example, these claims specify packaging client-initiated bi-directional protocol messages (e.g., TCP requests) into uni-directional protocol messages (e.g.,

HTTP messages) for transmission on the send channel (Claim 1, lines 14 - 16), and packaging server-initiated bi-directional protocol messages (e.g., TCP requests) into uni-directional protocol messages (e.g., HTTP messages) for transmission on the receive channel (Claim 1, lines 17 - 19).

2. The send channel is established from a first component on a client side of a network, through one or more uni-directional (e.g., HTTP-based) systems, to a second component on a remote side of the network (Claim 1, lines 4 - 6). The receive channel is also established from this same first component to the same second component (Claim 1, lines 7 - 9). Specification, p. 16, lines 12 - 16. Messages transmitted on the send and receive channels use a uni-directional protocol, such as HTTP (Claim 1, lines 14 - 19). By contrast, a bi-directional protocol (such as TCP) is used for a "first" connection from the client to the first component (Claim 1, lines 10 - 11) as well as for a "second" connection from the second component to the server (Claim 1, lines 12 - 13). See **Fig. 3**, where the first connection uses reference number **310** and the second connection uses reference number **370**. In a TCP environment, the client and server can then exchange TCP messages over an HTTP network such as the Internet, without requiring TCP ports to be opened through a firewall or HTTP proxy. (Drawbacks of prior art techniques in this environment are discussed in Appellant's specification on p. 3, line 9 - p. 4, line 13.)

3. Independent Claims 1 and 10 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellant's specification, as will now be described. Page 14, lines 8 - 9 and 13 - 17; **Fig. 3**, reference number **320**; p. 16, lines 12 - 16; and p. 17, lines 21 - 21 describe "means for establishing a send channel ...". Page 14, lines

9 - 13 and 17 - 19; **Fig. 3**, reference number **360**; p. 16, lines 12 - 16; and p. 18, lines 12 - 14 describe “means for establishing a receive channel ...”. Page 16, lines 4 - 7 and 9 - 11 and **Fig. 3**, reference number **310** describe “means for establishing a first TCP connection ...”. Page 16, lines 7 - 11; p. 18, lines 10 - 12; and **Fig. 3**, reference number **370** describe “means for establishing a second TCP connection ...”. Page 17, lines 8 - 19; p. 18, lines 1 - 6; and **Fig. 4A** describe “means for transmitting client-initiated TCP requests ... on the send channel”. Page 20, line 1 - p. 21, line 5 and **Fig. 4B** describe “means for transmitting server-initiated TCP requests ... on the receive channel”.

4. Dependent Claims 2 and 11 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants’ specification, as will now be described. Page 17, lines 8 - 12 describe “means for receiving a client-initiated TCP request ...”. In **Fig. 4A**, reference number **301** represents “the client”, reference number **310** represents “the first TCP connection”, reference number **320** represents “the first component”, and reference number **400** represents “a client-initiated TCP request”. Page 17, lines 13 - 18 describe “means for packaging the received client-initiated TCP request ...”, and reference number **410** of **Fig. 4A** illustrates the “HTTP POST request message” into which the TCP request is packaged. Page 17, lines 18 - 21 describe “means for sending the HTTP POST request message ... on the send channel”; see also reference numbers **360** (“the second component”), **340** (“the send channel”) and **410** (“the HTTP POST request message”) of **Fig. 4A**. Page 18, lines 1 - 3 and reference number **360** of **Fig. 4A** describe “means for receiving the sent HTTP POST request message ...”. Page 18, line 4 and “message conversion” (second occurrence) in **Fig. 4A**

describe "means for extracting ...". Page 18, line 5 - 6 describe "means for forwarding ..."; see also reference number **381** ("to the target server"), **370** ("on the second TCP connection"), and **420** ("the extracted client-initiated TCP request") of **Fig. 4A**.

5. Dependent Claims 3 and 12 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 18, lines 6 - 7 and are shown by reference number **430** in **Fig. 4A**.

6. Dependent Claims 4 and 13 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 17, lines 20 - 21 and p. 18, lines 7 - 9 ("... means for establishing the send channel operates in response to ..."); p. 18, lines 6 - 7 ("means for receiving the HTTP POST response ..."); and p. 18, lines 7 - 8 ("means for closing the send channel ...").

7. Dependent Claims 5 and 14 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification, as will now be described. Page 20, lines 1 - 5 describe "means for sending an HTTP GET request message ... on the receive channel"; see also **Fig. 4A**, reference numbers **460** ("an HTTP GET request message"), **320** ("the first component"), **360** ("the second component"), and **330** ("the receive channel"). Page 20, lines 10 - 11 describe "means for receiving the sent HTTP GET request message ...". Page 20, lines 12 - 15 describe "means for receiving a server-initiated TCP request ..."; see also **Fig. 4A**, reference numbers **470** ("a server-initiated TCP request"); **381** ("the

target server"); **360** ("the second component"); and **370** ("the second TCP connection"). Page 20, lines 16 - 18 and 21 describe "means for packaging ...". Page 20, lines 18 - 19 describe "means for sending the HTTP GET response message ... on the receive channel"; see also **Fig. 4A**, reference numbers **480** ("the HTTP GET response message"); **360** ("the second component"); **320** ("the first component"); and **330** ("the receive channel"). Page 20, lines 20 - 21 describe "means for receiving the sent HTTP GET response message ...". Page 21, lines 1 - 2 describe "means for extracting ...". Page 21, lines 2 - 3 describe "means for forwarding ..."; see also **Fig. 4A**, reference numbers **470** ("the extracted server-initiated TCP request"); **301** ("the client"); and **310** ("the first TCP connection").

8. Dependent Claims 6 and 15 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 20, lines 10 - 12 ("means for performing a read operation ...") and p. 20, lines 12 - 16 ("means for using the received server-initiated TCP request ...").

9. Dependent Claims 7 and 16 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification on p. 20, lines 21 - 22 ("preparing to receive another server-initiated TCP request ...").

6) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

10. The first ground of rejection presented for review is a rejection of Claims 1 - 4, 6 - 13, 15 - 22, 24 - 29, and 31 under 35 U.S.C. §103(a), citing U. S. Patent 6,412,009 to Erickson et al. and

U. S. Patent 6,442,590 to Inala et al.

11. The second ground of rejection presented for review is a rejection of Claims 5, 14, 23, and 30 under 35 U.S.C. §103(a), citing Erickson and Inala and further in view of RFC 2068 ("Fielding").

7) ARGUMENT

7.1) First Ground of Rejection

12. Paragraph 1 of the Office Action dated March 25, 2005 (hereinafter, "the Office Action") states that Claims 1 - 4, 6 - 13, 15 - 22, 24 - 29, and 31 are rejected under 35 U.S.C. §103(a) as being unpatentable over U. S. Patent 6,412,009 to Erickson et al. (hereinafter, "Erickson") in view of U. S. Patent 6,442,590 to Inala et al. (hereinafter, "Inala"). Of these claims, the independent claims are 1, 10, 19, 28, and 31.

13. Appellant respectfully submits that a *prima facie* case of obviousness under 35 U.S.C. §103 has not been made out as to his Claims 1 - 4, 6 - 13, 15 - 22, 24 - 29, and 31. Section 706.02(j) of the MPEP, "Contents of a 35 U.S.C. 103 Rejection", states the requirements for establishing a *prima facie* case of obviousness under this statute, noting that three criteria must be met. These criteria are (1) a suggestion or motivation, found either in the references or in the knowledge generally available, to modify or combine the references; (2) a reasonable expectation of success; and (3) the combination must teach or suggest all the claim limitations. This text goes on to state that "The initial burden is on the examiner to provide some suggestion of the

desirability of doing what the inventor has done.". The three requirements for establishing a *prima facie* case of obviousness are also stated in MPEP §2142, "Legal Concept of *Prima Facie* Obviousness", and MPEP §2143, "Basic Requirements of a *Prima Facie* Case of Obviousness".

7.1.1) Rejection of Independent Claims 1, 10, 19, 28, and 31

14. The §103 rejection of independent Claims 1, 10, 19, 28, and 31 is defective in several ways. First, limitations of Appellant's claims have been incorrectly evaluated, and thus limitations exist which are not taught by the references or combinations thereof. In fact, Appellant respectfully submits that Erickson teaches away from limitations of Appellant's independent claims. Finally, a proper motivation for combining the references has not been provided. These deficiencies in the §103 rejection will now be discussed in more detail.

15. Appellant respectfully disagrees with the Office Action's evaluation of the first and second elements of independent Claims 1, 10, 19, 28, and 31. In each claim, the first element specifies "... a send channel ..." and the second element specifies "... a receive channel ..., wherein the receive channel <u>is distinct from</u> the send channel" (emphasis added). Page 4, lines 1 - 2 of the Office Action admit that "Erickson does not <u>explicitly</u> teach the receive channel is distinct from the send channel" (emphasis added). Appellant respectfully submits that Erickson does not explicitly <u>or implicitly</u> teach use of distinct channels for sending and receiving, and in fact, that Erickson <u>explicitly</u> teaches use of a <u>single</u> tunnel/connection for <u>both</u> sending and receiving. Accordingly, Erickson <u>teaches away from</u> Appellant's use of a channel for sending that is distinct from a channel used for receiving. Citations to Erickson will now be discussed.

16. Reference is made (on p. 2, final line of the Office Action) to **Fig. 3** and col. 3, lines 3 - 29 of Erickson as teaching the "send channel" (i.e., first) element. What is taught in the cited text is "a persistent tunnel between a Web client and a Web server using an HTTP protocol for providing a persistent virtual connection between a host system and the Web client" (col. 3, lines 4 - 7).

17. Regarding Appellant's "receive channel" (i.e., second) element, p. 3, line 3 of the Office Action cites **Figs. 3 - 4**; col. 3, lines 3 - 29; and col. 7, line 63 - col. 8, line 4. However, neither **Fig. 3** nor **Fig. 4** illustrates two <u>distinct</u> channels between Erickson's Web client **126** and Web server **120**. **Figs. 3 - 4** therefore fail to teach the second, "receive channel", element of Appellant's independent claims.

18. Col. 3, lines 3 - 29 also fails to teach a <u>distinct</u> channel, beyond the "persistent tunnel between a Web client [**126**] and a Web server [**120**]" that was discussed above in paragraph 16. Because the Office Action has used this "persistent tunnel" for teaching Appellant's "send channel", the persistent tunnel cannot therefore use for teaching Appellant's "receive channel", which is explicitly specified in Appellant's independent claims as being "distinct from" the send channel.

19. In addition, col. 3, lines 12 - 16 of the cited text from Erickson teaches using "the connection" (see line 16) for messages sent to the Web client (i.e., server-initiated messages), and lines 18 - 23 teach using <u>this same connection</u> (i.e., "the connection"; see line 23) for

messages transmitted "[i]n the other direction" (i.e., client-initiated messages; see line 18). In other words, the cited text in col. 3, lines 3 - 29 teaches using a single connection for transmissions in both directions, in contrast to Appellant's claimed approach of using distinct channels for each direction.

20. The cited text in col. 7, line 63 - col. 8, line 4 discusses passing messages from Web server software **121** to a Web server extension **[132]** "without waiting for additional chunked messages". Appellant respectfully submits that this text is not relevant to a receive channel established between a client-side component and a server-side component, as is specified in the second element of Appellant's independent claims, and in particular, that this cited text fails to teach a receive channel that is "distinct from the send channel".

21. A number of additional references can be found in Erickson which indicate that Erickson teaches use of a single connection for transmitting messages between his Web client and Web server, in both directions, where the messages in opposing directions are interleaved on the single connection (in contrast to Appellant's use of distinct channels for each direction). See, for example, the final sentence of the Abstract ("The bi-directional persistent connection allows interleaving of the chunked data messages from the Web client with the chunked data messages on [presumably, "of" or "from"] the Web server on the [note, singular] persistent HTTP tunnel", emphasis added); **Fig. 3**, illustrating use of a single connection at HTTP tunnel **129**; col. 2, lines 40 - 59 ("... message is transmitted between a Web client and a Web server via an HTTP connection ...", lines 46 - 48, emphasis added, and "The chunked data messages from the Web

client are interleaved with the chunked data messages from the Web server on the [note, singular] persistent HTTP tunnel", lines 57 - 59, emphasis added); col. 3, lines 3 - 29 teach using one connection for transmissions in both directions, as discussed above in paragraphs 18 - 19; and col. 6, lines 28 - 35 ("This [i.e., not sending an "end chunk" message] allows one connection to remain active during the series of interleaved HTTP messages between the Web server and the Web client, thereby creating an HTTP tunnel **129** ...", emphasis added, and "The interleaved HTTP messages include messages that alternate between the Web client and the Web server as the sender [i.e., client-initiated messages and server-initiated messages] ...", emphasis added).

22. Thus, in summary, the Office Action fails to cite references that teach the second element of Appellants' independent Claims 1, 10, 19, 28, and 31, and in fact, teaches away from using a distinct channel.

23. Appellant also respectfully submits that the fifth and sixth limitations of his independent Claims 1, 10, 19, 28, and 31 have been incorrectly evaluated in the Office Action. In each claim, the fifth limitation pertains to use of the send channel for client-initiated requests, and the sixth limitation pertains to use of the receive channel for server-initiated requests. Page 3, lines 9 - 15 of the Office Action analyze the fifth limitation, and p. 3, lines 16 - 22 analyze the sixth limitation. Each of these will now be discussed.

24. Page 3, lines 9 - 15 of the Office Action cite col. 2, lines 43 - 48 and lines 41 - 59 of Erickson as teaching Appellant's fifth limitation where client-initiated requests "are transmitted

on the send channel". The cited text refers to "a persistent HTTP tunnel" (col. 2, line 41); "an HTTP connection" (col. 2, line 49) where this HTTP connection is used for data messages "transmitted between a Web client and a Web server" (col. 2, lines 47 - 49); and interleaving data messages "from the Web client ... with ... data messages from the Web server on the [note, singular] persistent HTTP tunnel" (col. 2, lines 57 - 59).

25. Page 3, lines 16 - 22 of the Office Action cite col. 2, lines 43 - 48 as well as col. 5, lines 53 - 58 and col. 7, lines 30 - 41 of Erickson as teaching Appellant's sixth limitation where server-initiated requests "are transmitted on the receive channel". The cited text in col. 5, lines 53 - 58 refers to communications between the Web server software **121** and Web server extension **132**. This text is analogous to that discussed above in paragraph 20, and as noted therein, is not pertinent to the receive channel specified in Appellant's claim language. The cited text in col. 7, lines 30 - 41 discusses server-initiated messages ("A typical exchange has the host system starting Telnet negotiation", col. 7, lines 31 - 33). However, lines 39 - 41 of this cited text specify that the server-initiated message "is sent via the HTTP tunnel **129** ... to the Web client". Notably, this HTTP tunnel **129** is the same tunnel used for transmitting messages in the opposite direction. See, for example, paragraph 21 above, discussing text from col. 6, lines 28 - 35 pertaining to interleaving messages where the message sender is the Web client with messages where the message sender is the Web server.

26. As has been discussed, Appellant's independent claims explicitly specify a receive channel that is distinct from a send channel. However, as demonstrated by paragraphs 24 - 25

above, the Office Action fails to cite references that teach use of a distinct receive channel for server-initiated requests, and accordingly, fails to cite references that teach the sixth element of Appellants' independent Claims 1, 10, 19, 28, and 31.

27. Because the Office Action fails to cite references that teach all of the limitations of the second and sixth elements of Appellants' independent Claims 1, 10, 19, 28, and 31, the §103 rejection violates the requirements of the above-noted Sections 706.02(j), 2142, and 2143 of the MPEP.

28. Furthermore, a proper motivation for combining the references has not been provided. As stated above, a proper motivation is a requirement for establishing a *prima facie* case of obviousness. Deficiencies in the supposed motivation to combine Erickson with Inala, presented on p. 4, lines 1 - 9 of the Office Action, to yield Appellant's independent claims will now be discussed.

29. As noted above in paragraph 15, the Office Action admits (on p. 4, lines 1 - 2) that Erickson does not "explicitly" teach a receive channel that is distinct from the send channel. Appellant has also demonstrated in paragraphs 16 - 26, above, that Erickson not only does not teach this limitation, but in fact, teaches away therefrom. See, for example, col. 7, lines 53 - 62, where Erickson teaches that use of his single connection provides performance improvements ("... Because only one connection is needed during the communication flow, the present invention provides performance [advantages]", emphasis added). (Col. 2, lines 2 - 10 and lines

24 - 32 discuss performance concerns of the prior art, and lines 32 - 33 state that the invention is "directed to filling this need [i.e., response time improvements]".) Accordingly, because Erickson teaches that only a single connection "is needed", and "the present [i.e., Erickson's] invention [thereby] provides performance [improvements]", Appellant respectfully submits that one of skill in the art would not be motivated to introduce multiple connections (as discussed in Inala) into Erickson's disclosed teachings.

30. Furthermore, Appellant respectfully submits that one of skill in the art would not be motivated to attempt a combination of Erickson and Inala because Inala is not directed to the environment in which Erickson's teachings operate. Erickson's environment includes a Web client **126** and an emulator **134**, communicating with a host system **110** that is connected via a LAN **116** to a Web server **120**. The host system and emulator are described as using connection-oriented session data, such as Telnet (col. 4, lines 13 - 15), and encapsulating the connection-oriented session data for transmission using HTTP (col. 4, lines 13 - 16). Inala has no teaching of such a "mixed protocol" environment.

31. Page 4, line 4 of the Office Action cites col. 8, lines 30 - 32 of Inala as teaching use of two distinct channels. In fact, Inala does state that one HTTP connection can be used for sending data and one (separate) HTTP connection can be used for receiving data. However, this is unrelated to use of distinct channels where TCP (or Telnet, as in Erickson) requests are packaged into HTTP messages (as in Appellant's independent Claims 1, 10, and 19) or where bi-directional protocol requests are packaged into uni-directional protocol messages (as in

Appellant's independent Claims 28 and 31). Instead, the cited text from Inala merely teaches that two HTTP connections can be "opened to server **17** for the purpose", emphasis added, where "the purpose" is apparently a reference to col. 8, lines 25 - 29, stating that [client] software **37** uses HTTP to communicate with, and receive data from, "a main Internet service control" server **17** (see col. 5, lines 45 - 46, providing this characterization of server **17**).

32. Furthermore, Appellant respectfully submits that one of skill in the art would not be motivated to consult Inala's teachings when searching for a solution to the problem to which Appellant's invention is directed (namely, exchanging bi-directional data over a uni-directional network). Inala's teachings pertain to interactive chat sessions and advertising during such sessions. See, for example, lines 4 - 5 and 9 - 10 of the Abstract; col. 2, line 66 - col. 3, line 2; and col. 4, lines 54 - 61. Opening two connections between client software and a server for ad-based chat sessions (as in Inala) is not relevant to exchanging bi-directional data over a uni-directional network (as in Appellant's invention).

33. Furthermore, Appellant respectfully submits that one of skill in the art would fail to locate Inala's teachings absent use of hindsight reconstruction. That is, Appellant finds nothing in Inala's teachings that is relevant to his claimed invention except the phrase "two HTTP connections" and the statement that "One connection is for sending data and one is for receiving data.". This suggests use of a search engine, into which terms from Appellant's teachings were inserted, in order to pick and choose references in an attempt to recreate Appellant's claimed invention. This hindsight reconstruction approach is prohibited in an obviousness analysis. See,

for example, the holding in ***In re Fritch***, 23 USPQ 2d 1780, 1784 (Fed. Cir. 1992), which stated

> It is impermissible to use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. This court has previously stated that "[o]ne cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention." (quoting *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ 2d 1596, 1600 (Fed. Cir. 1988)).."

34. In addition, Appellant respectfully disagrees with the statements on p. 4, lines 4 - 9, which state that one of skill in the art would have used Inala's two distinct channels with Erickson's teachings because "This [use of two channels] would have improved the efficiency of transmission in term [sic] of cost and simplicity required for the connections.". Appellant fails to see how changing from Erickson's single connection to use of two channels, according to the supposed motivation stated in the Office Action, would <u>in any way</u> reduce cost and/or simplicity, or improve the efficiency, of Erickson's disclosed approach. Accordingly, Appellant respectfully submits that the supposed motivation is a mere unfounded assertion, with no basis in fact therefor having been provided.

35. As demonstrated above by paragraphs 14 - 34, Appellant respectfully submits that a proper §103 rejection has not been made out as to his independent Claims 1, 10, 19, 28, and 31. Without more, these claims are deemed patentable. See ***In re Oetiker***, 24 USPQ 2d 1443, 1444 (Fed. Cir. 1992), which stated:

> If the examination at the initial stage does not produce a prima facie case of unpatentability, then without more the applicant is entitled to grant of the patent.

7.1.2) Rejection of Dependent Claims 2 - 3, 11 - 12, 20 - 21, and 29

36. Dependent Claims 2 - 3, 11 - 12, 20 - 21, and 29 specify further details pertaining to transmission of client-initiated requests on the send channel.

37. Having failed to establish that independent Claims 1, 10, 19, and 28 are obvious, the rejection of dependent Claims 2 - 3, 11 - 12, 20 - 21, and 29 fails as well. See §2143.03 of the MPEP, which states that

> If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious.

In re Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Dependent Claims 2 - 3, 11 - 12, 20 - 21, and 29 are therefore deemed nonobvious.

7.1.3) Rejection of Dependent Claims 4, 13, and 22

38. Dependent Claims 4, 13, and 22 specify limitations pertaining to establishing, and closing, the send channel. Appellant respectfully submits that the Office Action evaluation of the final limitation of these claims, which is directed to the closing of the send channel (i.e., "closing the send channel, responsive to ... receiving the HTTP POST response"), is improper. Page 5, lines 5 - 9 of the Office Action -- an in particular, lines 8 - 9 thereof -- cites col. 2, lines 11 - 15 of Erickson as teaching this final limitation.

39. The cited text in col. 2 pertains to prior art techniques, stating that "After the Web server sends a response to the browser **25**, the connection is closed.". Appellant is entitled to have all words of his claim language considered in an obviousness analysis. Neither the prior art nor

Erickson teach closing a send channel, responsive to receiving an HTTP POST response, in the content of using a send channel that is distinct from a receive channel (as claimed in Appellant's independent claims).

40. Because Erickson fails to teach this limitation, the Office Action fails to make out a *prima facie* case of obviousness as to dependent Claims 4, 13, and 22. (Inala also fails to teach this limitation.)

41. Because a *prima facie* case of obviousness has not been made out as to dependent Claims 4, 13, and 22, as demonstrated above by paragraphs 38 - 40, these claims are deemed patentable, according to the holding in the above-quoted ***In re Oetiker***.

42. In addition, having failed to establish that independent Claims 1, 10, and 19 are obvious, the rejection of their dependent Claims 4, 13, and 22 fails as well, according to the above-cited MPEP §2143.03 and the above-quoted ***In re Fine***. Thus, dependent Claims 4, 13, and 22 are deemed nonobvious for this reason as well.

7.1.4) Rejection of Dependent Claims 6, 15, and 24

43. Dependent Claims 6, 15, and 24 specify further details pertaining to transmission of server-initiated requests on the receive channel. In particular, these claims specify a first limitation of "performing a read operation on the second TCP connection, <u>responsive to</u> ... receiving the sent HTTP GET request message <u>and prior to</u> ... receiving the server-initiated TCP

request" (emphasis added) and a second limitation of "using the received server-initiated TCP request as a result of the read operation, thereby triggering ... packaging [of] the receiver server-initiated TCP request in the HTTP GET response message" (emphasis added). Page 5, lines 10 - 12 of the Office Action cite col. 7, lines 30 - 41 of Erickson as teaching these limitations.

44. However, the cited text simply discusses an exchange started by a host system, and states that the data received from that exchange is embedded in a chunk for sending over the HTTP tunnel **129** to the Web client. There is no discussion therein of a "read operation", or performing such read operation "responsive to ... receiving the sent HTTP GET request message and prior to ... receiving the server-initiated TCP request". In fact, the Office Action admits (p. 7, lines 10 - 11) that Erickson and Inala do not [explicitly] teach HTTP GET request messages (and Appellant respectfully submits that Erickson teaches only use of HTTP POST messages). Accordingly, the Office Action fails to make out a *prima facie* case of obviousness as to dependent Claims 6, 15, and 24.

45. Because a *prima facie* case of obviousness has not been made out as to their dependent Claims 6, 15, and 24, as demonstrated above by paragraphs 43 - 44, these claims are deemed patentable, according to the holding in the above-quoted ***In re Oetiker***.

46. In addition, having failed to establish that independent Claims 1, 10, and 19 are obvious, the rejection of their dependent Claims 6, 15, and 24 fails as well, according to the above-cited MPEP §2143.03 and the above-quoted ***In re Fine***. Thus, dependent Claims 6, 15, and 24 are

deemed nonobvious for this reason as well.

7.1.5) Rejection of Dependent Claims 7, 16, and 25

47. Dependent Claims 7, 16, and 25 specify further details pertaining to transmission of server-initiated requests on the receive channel. In particular, these claims specify a limitation of "preparing to receive another server-initiated TCP request by triggering ... sending [of] the HTTP GET request message ...". That is, an HTTP GET request message is sent, responsive to receiving an HTTP GET response message, in order to prepare for receiving the next server-initiated request. (See Appellant's specification, p. 20, lines 20 - 22.) Page 5, lines 13 - 15 of the Office Action cite col. 10, line 4 - 5 and col. 7, lines 3 - 13 of Erickson as teaching this limitation.

48. However, the analysis in the Office Action has apparently overlooked the claim language "preparing to receive another server-initiated TCP request by triggering operation of ..." (emphasis added), as the cited text in col. 10, lines 4 - 5 is not relevant thereto. Furthermore, the cited text from col. 7, lines 3 - 13 discusses only HTTP POST messages, and is therefore irrelevant to Appellant's claim language, which refers to an HTTP GET response message. Accordingly, the Office Action fails to make out a *prima facie* case of obviousness as to dependent Claims 7, 16, and 25.

49. Because a *prima facie* case of obviousness has not been made out as to their dependent Claims 7, 16, and 25, as demonstrated above by paragraphs 47 - 48, these claims are deemed

patentable, according to the holding in the above-quoted ***In re Oetiker***.

50. In addition, having failed to establish that independent Claims 1, 10, and 19 are obvious, the rejection of their dependent Claims 7, 16, and 25 fails as well, according to the above-cited MPEP §2143.03 and the above-quoted ***In re Fine***. Thus, dependent Claims 7, 16, and 25 are deemed nonobvious for this reason as well.

7.1.6) Rejection of Dependent Claims 8 - 9, 17 - 18, and 26 - 27

51. Dependent Claims 8 - 9, 17 - 18, and 26 - 27 specify particular MIME types for HTTP POST and HTTP GET request messages.

52. Having failed to establish that independent Claims 1, 10, and 19 are obvious, the rejection of their dependent Claims 8 - 9, 17 - 18, and 26 - 27 fails as well, according to the above-cited MPEP §2143.03 and the above-quoted ***In re Fine***. Thus, dependent Claims 8 - 9, 17 - 18, and 26 - 27 are deemed nonobvious.

7.2) Second Ground of Rejection

53. Paragraph 2 of the Office Action states that Claims 5, 14, 23, and 30 are rejected under 35 U.S.C. §103(a), citing Erickson and Inala and further in view of RFC 2068 ("Fielding"). These claims are all dependent claims.

54. Appellant respectfully submits that the §103 rejection of dependent Claims 5, 14, 23, and

30 is defective. Limitations of Appellant's claims have been incorrectly evaluated, and thus limitations exist which are not taught by the references or combinations thereof; accordingly, Appellant respectfully submits that a *prima facie* case of obviousness under 35 U.S.C. §103 (requirements for which were discussed above in paragraph 13) has not been made out as to Claims 5, 14, 23, and 30, as will now be discussed.

7.2.1) Rejection of Dependent Claims 5, 14, and 23

55. Dependent Claims 5, 14, and 23 specify details pertaining to transmission of server-initiated requests on the receive channel. In particular, each limitation of Claims 5, 14, and 23, except the third and eighth limitations, refers to "HTTP GET request" or "HTTP GET response" messages.

56. When analyzing these dependent claims, the Office Action admits that Erickson and Inala fail to [explicitly] teach HTTP GET requests. See p. 7, lines 10 - 11 of the Office Action. The Office Action then continues by citing Fielding, and states that "it would have been obvious ... to modify Erickson-Inala in view of Fielding because such GET request would allow to retrieve only information identified by the Request-URI", and that this "would have reduced unnecessary network usage". (See p. 7, lines 12 - 16 of the Office Action.) Appellant respectfully submits that this supposed motivation for combining the references is flawed. The claims at issue pertain to server-initiated request messages. When the server is initiating the communication, there is no "Request-URI" with which information is being requested from the server (because if there was, then the server would be responding to a client-initiated communication). Accordingly,

Appellant respectfully submits that a proper motivation to combine the references is missing, and the Office Action therefore fails to make out a *prima facie* case of obviousness as to dependent Claims 5, 14, and 23.

57. Because a *prima facie* case of obviousness has not been made out as to their dependent Claims 5, 14, and 23, as demonstrated above by paragraphs 56 - 57, these claims are deemed patentable, according to the holding in the above-quoted ***In re Oetiker***.

58. In addition, having failed to establish that independent Claims 1, 10, and 19 are obvious, the rejection of their dependent Claims 5, 14, and 23 fails as well, according to the above-cited MPEP §2143.03 and the above-quoted ***In re Fine***. Thus, dependent Claims 5, 14, and 23 are deemed nonobvious for this reason as well.

7.2.2) Rejection of Dependent Claim 30

59. Dependent Claim 30 specifies details pertaining to transmission of server-initiated requests on the receive channel, and is similar to dependent Claims 5, 14, and 23. Each limitation of Claim 30, except the third and eighth, refers to "read request" or "read response" messages.

60. The Office Action has failed to provide an independent evaluation of Claim 30, even though it contains claim language which differs from that of Claims 5, 14, and 23. Accordingly, Appellant respectfully submits that the Office Action fails to make out a *prima facie* case of

obviousness as to dependent Claim 30.

61. Because a *prima facie* case of obviousness has not been made out as to dependent Claim 30, as demonstrated above by paragraphs 59 - 60, this claim is deemed patentable, according to the holding in the above-quoted ***In re Oetiker***.

62. In addition, having failed to establish that independent Claims 28 is obvious, the rejection of its dependent Claim 30 fails as well, according to the above-cited MPEP §2143.03 and the above-quoted ***In re Fine***. Thus, dependent Claim 30 is deemed nonobvious for this reason as well.

8) CONCLUSION

For the reasons set out above, Appellant respectfully contends that each appealed claim is patentable, and respectfully requests that Examiner's Final Rejection of appealed Claims 1 - 31 should be reversed.

Respectfully submitted,

Marcia L. Doubet

Marcia L. Doubet
Attorney for Appellant
Reg. No. 40,999

Customer Number for Correspondence: 43168
Phone: 407-343-7586
Fax: 407-343-7587

CLAIMS APPENDIX

CLAIMS AS CURRENTLY PRESENTED:

1. A computer program product for sending Transmission Control Protocol (TCP) messages through HyperText Transfer Protocol (HTTP) systems, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code means for establishing a send channel from a first component on a client side of a network, through one or more HTTP-based systems, to a second component on a remote side of the network;

computer-readable program code means for establishing a receive channel from the first component, through the one or more HTTP-based systems, to the second component, wherein the receive channel is distinct from the send channel;

computer-readable program code means for establishing a first TCP connection from a client on the client side to the first component;

computer-readable program code means for establishing a second TCP connection from the second component to a target server on the remote side;

computer-readable program code means for transmitting client-initiated TCP requests from the client to the target server by packaging the client-initiated TCP requests into HTTP messages which are transmitted on the send channel; and

computer-readable program code means for transmitting server-initiated TCP requests from the target server to the client by packaging the server-initiated TCP requests into HTTP messages which are transmitted on the receive channel.

2. The computer program product according to Claim 1, wherein the computer-readable program code means for transmitting client-initiated TCP requests further comprises:

computer-readable program code means for receiving a client-initiated TCP request from the client at the first component on the first TCP connection;

computer-readable program code means for packaging the received client-initiated TCP request in an HTTP POST request message;

computer-readable program code means for sending the HTTP POST request message to the second component on the send channel;

computer-readable program code means for receiving the sent HTTP POST request message at the second component;

computer-readable program code means for extracting the client-initiated TCP request from the received HTTP POST request message; and

computer-readable program code means for forwarding the extracted client-initiated TCP request to the target server on the second TCP connection.

3. The computer program product according to Claim 2, wherein the computer-readable program code means for transmitting client-initiated TCP requests further comprises computer-readable program code means for acknowledging the HTTP POST request by sending an HTTP POST response from the second component to the first component on the send channel.

4. The computer program product according to Claim 3, wherein the computer-readable program code means for establishing the send channel operates in response to the computer-readable

program code means for receiving the client-initiated TCP request, and wherein the computer-readable program code means for transmitting client-initiated TCP requests further comprises:

computer-readable program code means for receiving the HTTP POST response at the first component; and

computer-readable program code means for closing the send channel, responsive to operation of the computer-readable program code means for receiving the HTTP POST response.

5. The computer program product according to Claim 1, wherein the computer-readable program code means for transmitting server-initiated TCP requests further comprises:

computer-readable program code means for sending an HTTP GET request message from the first component to the second component on the receive channel;

computer-readable program code means for receiving the sent HTTP GET request message at the second component;

computer-readable program code means for receiving a server-initiated TCP request from the target server at the second component on the second TCP connection;

computer-readable program code means for packaging the received server-initiated TCP request in an HTTP GET response message which acknowledges the received HTTP GET request message;

computer-readable program code means for sending the HTTP GET response message from the second component to the first component on the receive channel;

computer-readable program code means for receiving the sent HTTP GET response message at the first component;

computer-readable program code means for extracting the server-initiated TCP request from the received HTTP GET response message; and

computer-readable program code means for forwarding the extracted server-initiated TCP request to the client on the first TCP connection.

6. The computer program product according to Claim 5, wherein the computer-readable program code means for transmitting server-initiated TCP requests further comprises:

computer-readable program code means for performing a read operation on the second TCP connection, responsive to operation of the computer-readable program code means for receiving the sent HTTP GET request message and prior to operation of the computer-readable program code means for receiving the server-initiated TCP request; and

computer-readable program code means for using the received server-initiated TCP request as a result of the read operation, thereby triggering operation of the computer-readable program code means for packaging the received server-initiated TCP request in the HTTP GET response message.

7. The computer program product according to Claim 5, wherein the computer-readable program code means for transmitting server-initiated TCP requests further comprises computer-readable program code means for preparing to receive another server-initiated TCP request by triggering operation of the computer-readable program code means for sending the HTTP GET request message from the first component to the second component, responsive to operation of the computer-readable program code means for receiving the sent HTTP GET response message at

the first component.

8. The computer program product according to Claim 2, wherein a Multi-Purpose Internet Mail Extensions (MIME) type of the HTTP POST request message is set to “binary/tcp”.

9. The computer program product according to Claim 5, wherein a Multi-Purpose Internet Mail Extensions (MIME) type of the HTTP GET request message is set to “binary/tcp”.

10. A system for sending Transmission Control Protocol (TCP) messages through HyperText Transfer Protocol (HTTP) systems, comprising:

means for establishing a send channel from a first component on a client side of a network, through one or more HTTP-based systems, to a second component on a remote side of the network;

means for establishing a receive channel from the first component, through the one or more HTTP-based systems, to the second component, wherein the receive channel is distinct from the send channel;

means for establishing a first TCP connection from a client on the client side to the first component;

means for establishing a second TCP connection from the second component to a target server on the remote side;

means for transmitting client-initiated TCP requests from the client to the target server by packaging the client-initiated requests into HTTP messages which are transmitted on the send channel; and

means for transmitting server-initiated TCP requests from the target server to the client by packaging the server-initiated requests into HTTP messages which are transmitted on the receive channel.

11. The system according to Claim 10, wherein the means for transmitting client-initiated TCP requests further comprises:

means for receiving a client-initiated TCP request from the client at the first component on the first TCP connection;

means for packaging the received client-initiated TCP request in an HTTP POST request message;

means for sending the HTTP POST request message to the second component on the send channel;

means for receiving the sent HTTP POST request message at the second component;

means for extracting the client-initiated TCP request from the received HTTP POST request message; and

means for forwarding the extracted client-initiated TCP request to the target server on the second TCP connection.

12. The system according to Claim 11, wherein the means for transmitting client-initiated TCP requests further comprises means for acknowledging the HTTP POST request by sending an HTTP POST response from the second component to the first component on the send channel.

13. The system according to Claim 12, wherein the means for establishing the send channel operates in response to the means for receiving the client-initiated TCP request, and wherein the means for transmitting client-initiated TCP requests further comprises:

means for receiving the HTTP POST response at the first component; and

means for closing the send channel, responsive to operation of the means for receiving the HTTP POST response.

14. The system according to Claim 10, wherein the means for transmitting server-initiated TCP requests further comprises:

means for sending an HTTP GET request message from the first component to the second component on the receive channel;

means for receiving the sent HTTP GET request message at the second component;

means for receiving a server-initiated TCP request from the target server at the second component on the second TCP connection;

means for packaging the received server-initiated TCP request in an HTTP GET response message which acknowledges the received HTTP GET request message;

means for sending the HTTP GET response message from the second component to the first component on the receive channel;

means for receiving the sent HTTP GET response message at the first component;

means for extracting the server-initiated TCP request from the received HTTP GET response message; and means for forwarding the extracted server-initiated TCP request to the client on the first TCP connection.

15. The system according to Claim 14, wherein the means for transmitting server-initiated TCP requests further comprises:

means for performing a read operation on the second TCP connection, responsive to operation of the means for receiving the sent HTTP GET request message and prior to operation of the means for receiving the server-initiated TCP request; and

means for using the received server-initiated TCP request as a result of the read operation, thereby triggering operation of the means for packaging the received server-initiated TCP request in the HTTP GET response message.

16. The system according to Claim 14, wherein the means for transmitting server-initiated TCP requests further comprises means for preparing to receive another server-initiated TCP request by triggering operation of the means for sending the HTTP GET request message from the first component to the second component, responsive to operation of the means for receiving the sent HTTP GET response message at the first component.

17. The system according to Claim 11, wherein a Multi-Purpose Internet Mail Extensions (MIME) type of the HTTP POST request message is set to "binary/tcp".

18. The system according to Claim 14, wherein a Multi-Purpose Internet Mail Extensions

(MIME) type of the HTTP GET request message is set to "binary/tcp".

19. A method for sending Transmission Control Protocol (TCP) messages through HyperText Transfer Protocol (HTTP) systems, comprising the steps of:

establishing a send channel from a first component on a client side of a network , through one or more HTTP-based systems, to a second component on a remote side of the network;

establishing a receive channel from the first component, through the one or more HTTP-based systems, to the second component, wherein the receive channel is distinct from the send channel;

establishing a first TCP connection from a client on the client side to the first component;

establishing a second TCP connection from the second component to a target server on the remote side;

transmitting client-initiated TCP requests from the client to the target server by packaging the client-initiated requests into HTTP messages which are transmitted on the send channel; and

transmitting server-initiated TCP requests from the target server to the client by packaging the server-initiated requests into HTTP messages which are transmitted on the receive channel.

20. The method according to Claim 19, wherein the step of transmitting client-initiated TCP requests further comprises the steps of:

receiving a client-initiated TCP request from the client at the first component on the first TCP connection;

packaging the received client-initiated TCP request in an HTTP POST request message;

sending the HTTP POST request message to the second component on the send channel;

receiving the sent HTTP POST request message at the second component;

extracting the client-initiated TCP request from the received HTTP POST request message; and

forwarding the extracted client-initiated TCP request to the target server on the second TCP connection.

21. The method according to Claim 20, wherein the step of transmitting client-initiated TCP requests further comprises the step of acknowledging the HTTP POST request by sending an HTTP POST response from the second component to the first component on the send channel.

22. The method according to Claim 21, wherein the step of establishing the send channel operates in response to the step of receiving the client-initiated TCP request, and wherein the step of transmitting client-initiated TCP requests further comprises the steps of:

receiving the HTTP POST response at the first component; and

closing the send channel, responsive to receiving the HTTP POST response.

23. The method according to Claim 19, wherein the step of transmitting server-initiated TCP requests further comprises the steps of:

sending an HTTP GET request message from the first component to the second component on the receive channel;

receiving the sent HTTP GET request message at the second component;

receiving a server-initiated TCP request from the target server at the second component on the second TCP connection;

packaging the received server-initiated TCP request in an HTTP GET response message which acknowledges the received HTTP GET request message;

sending the HTTP GET response message from the second component to the first component on the receive channel;

receiving the sent HTTP GET response message at the first component;

extracting the server-initiated TCP request from the received HTTP GET response message; and

forwarding the extracted server-initiated TCP request to the client on the first TCP connection.

24. The method according to Claim 23, wherein the step of transmitting server-initiated TCP requests further comprises the steps of:

performing a read operation on the second TCP connection, responsive to receiving the sent HTTP GET request message and prior to receiving the server-initiated TCP request; and

using the received server-initiated TCP request as a result of the read operation, thereby triggering the step of packaging the received server-initiated TCP request in the HTTP GET response message.

25. The method according to Claim 23, wherein the step of transmitting server-initiated TCP

requests further comprises the step of preparing to receive another server-initiated TCP request by triggering the step of sending the HTTP GET request message from the first component to the second component, responsive to receiving the sent HTTP GET response message at the first component.

26. The method according to Claim 20, wherein a Multi-Purpose Internet Mail Extensions (MIME) type of the HTTP POST request message is set to “binary/tcp”.

27. The method according to Claim 23, wherein a Multi-Purpose Internet Mail Extensions (MIME) type of the HTTP GET request message is set to “binary/tcp”.

28. A method for transporting bi-directional protocol traffic through uni-directional protocol systems, comprising the steps of:

establishing a send channel from a first component on a client side of a network connection, through one or more uni-directional protocol-based systems, to a second component on a remote side of the network connection;

establishing a receive channel from the first component, through the one or more uni-directional protocol-based systems, to the second component, wherein the receive channel is distinct from the send channel;

establishing a first bi-directional protocol connection from a client on the client side to the first component;

establishing a second bi-directional protocol connection from the second component to a target server on the remote side;

transmitting client-initiated bi-directional protocol requests from the client to the target server by packaging the client-initiated bi-directional protocol requests into uni-directional protocol messages which are transmitted on the send channel; and

transmitting server-initiated bi-directional protocol requests from the target server to the client by packaging the server-initiated bi-directional protocol requests into uni-directional protocol messages which are transmitted on the receive channel.

29. The method according to Claim 28, wherein the step of transmitting client-initiated bi-directional protocol requests further comprises the steps of:

receiving a client-initiated bi-directional protocol request from the client at the first component on the first bi-directional protocol connection;

packaging the received client-initiated bi-directional protocol request in a uni-directional protocol write request message;

sending the uni-directional protocol write request message to the second component on the send channel;

receiving the sent uni-directional protocol write request message at the second component;

extracting the client-initiated bi-directional protocol request from the received uni-directional protocol write request message; and

forwarding the extracted client-initiated bi-directional protocol request to the target server on the second bi-directional protocol connection.

30. The method according to Claim 28, wherein the step of transmitting server-initiated bi-directional protocol requests further comprises the steps of:

sending a uni-directional protocol read request message from the first component to the second component on the receive channel;

receiving the sent uni-directional protocol read request message at the second component;

receiving a server-initiated bi-directional protocol request from the target server at the second component on the second bi-directional protocol connection;

packaging the received server-initiated bi-directional protocol request in a uni-directional protocol read response message which acknowledges the received uni-directional protocol read request message;

sending the uni-directional protocol read response message from the second component to the first component on the receive channel;

receiving the sent uni-directional protocol read response message at the first component;

extracting the server-initiated bi-directional protocol request from the received uni-directional protocol read response message; and

forwarding the extracted server-initiated bi-directional protocol request to the client on the first bi-directional protocol connection.

31. A system for providing bi-directional messaging over uni-directional protocol systems, comprising:

a send channel established from a first component on a client side of a network connection, through at least one uni-directional protocol-based system, to a second component on a remote side of the network connection;

a receive channel established from the first component, through the at least one uni-directional protocol-based system, to the second component, wherein the receive channel is distinct from the send channel;

a first bi-directional protocol connection established between a client on the client side and the first component; and

a second bi-directional protocol connection established between the second component and a server on the remote side;

wherein the first component packages client-initiated bi-directional protocol requests, which are sent from the client on the first bi-directional protocol connection and received at the first component, into uni-directional protocol messages and forwards the packaged client-initiated protocol requests to the second component using the send channel and upon receipt of the forwarded client-initiated requests, the second component extracts the client-initiated bi-directional protocol requests and forwards the extracted client-initiated bi-directional protocol requests to the server on the second bi-direction protocol connection, thereby providing client-to-server messaging through the at least one uni-directional protocol-based system; and

wherein the second component packages server-initiated bi-directional protocol requests, which are sent from the server on the second bi-directional protocol connection and received at the second component, into uni-directional protocol messages and forwards the packaged server-initiated protocol requests to the first component using the receive channel and upon receipt of the forwarded server-initiated requests, the first component extracts the server-initiated bi-directional protocol requests and forwards the extracted server-initiated bi-directional protocol requests to the client on the first bi-direction protocol connection, thereby providing server-to-client messaging through the at least one uni-directional protocol-based system.

EVIDENCE APPENDIX

Appellant, the Appellant's legal representative, and the assignee have no personal knowledge of evidence requiring separate identification herein as bearing on this Appeal.

RELATED PROCEEDINGS APPENDIX

No related proceedings are personally known to Appellant, the Appellant's legal representative, or the assignee.